**In the Claims:**

Claims 1-36 were previously pending.

Claims 13 and 28-30 are amended.

Claims 37-39 are added.

Claims 1-39 are pending.

**Listing of Claims:**

1. (Previously presented) In a computing device having a processor that generates a first address signal of a first width and one or more peripheral devices that are addressed with a second address signal of a second width that is greater than the first width, wherein the second address signal is produced in the computing device by concatenating an address extension from an address extension register with the first address signal, a method comprising:

concurrently executing threads of a plurality of application programs, wherein different ones of the threads indicate one or more address extensions to an operating system;

storing the address extensions for use by the operating system;

repeatedly switching between execution of the threads; and

prior to executing a particular thread, writing the address extension of a base address indicated by the particular thread to the extension register.

2. (Original) A method as recited in claim 1, wherein the address extensions are indicated as a value of the second width.



PAGE 4/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

3. (Original) A method as recited in claim 1, wherein individual address extensions identify address ranges associated with one or more peripheral devices.

4. (Previously presented) A method as recited in claim 1, further comprising calling an operating system device driver from one of the threads, wherein the device driver invokes an initialization function to indicate one or more base addresses.

5. (Original) A thread scheduler that schedules multiple execution threads for interleaved execution by a processor, wherein the processor generates a processor address signal that is combined with an extended address signal to create a peripheral address signal, wherein the extended address signal is produced from a value stored in an address extension register, and wherein the value is writeable and readable by the processor; the thread scheduler performing steps comprising:

interrupting a first execution thread to execute portions of one or more other execution threads;

recording the value from the address extension register;

restoring the recorded value to the extension register after executing said portions of one or more other threads; and

resuming the first execution thread after restoring the recorded value to the address extension register.

6. (Original) A thread scheduler as recited in claim 5, wherein the thread scheduler records address extension register values associated with a plurality of


PAGE 5/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

interrupted execution threads and restores the address extension register values to the address extension register when resuming the associated execution threads.

7. (Original) One or more computer-readable storage media containing a program that implements a thread scheduler as recited in claim 5.

8. (Original) A method of scheduling multiple execution threads for interleaved execution by a processor, wherein the processor generates a processor address signal, comprising:

executing a first execution thread;

writing an address extension to an extension register;

concatenating the address extension with the processor address signal to create a peripheral address signal used by the first execution thread;

storing the address extension in a location other than the extension register;

interrupting the first execution thread to execute portions of one or more other execution threads;

interrupting the one or more other execution threads to resume execution of the first execution thread;

restoring the stored address extension to the extension register before resuming executing of the first execution thread; and

resuming execution of the first execution thread.


PAGE 6/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

9. (Original) A method as recited in claim 8, wherein each address extension identifies an address range associated with one or more peripheral devices.

10. (Original) A method as recited in claim 8, further comprising:

writing a second address extension associated with the one or more other execution threads to the extension register;

concatenating the second address extension with a second processor address signal to create a second peripheral address signal used by the one or more other execution threads; and

storing the second address extension in a second location other than the extension register.

11. (Original) A computer-readable storage medium having instructions for performing the steps recited in claim 8.

12. (Original) A multi-tasking operating system for use in a computing device having a processor that generates a first address signal of a first width and one or more peripheral devices that are addressed with a second address signal of a second width that is greater than the first width, wherein the second address signal is produced in the computing device by concatenating an address extension from an address extension register with the first address signal, the operating system being configured to perform steps comprising:

concurrently executing a plurality of application program threads;


PAGE 7/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

storing address extensions corresponding to different ones of the application program threads;

repeatedly switching between the application program threads; and

prior to executing any particular thread, writing the address extension corresponding to that particular thread to the extension register.

b1

13. (Currently amended) A multi-tasking operating system as recited in claim 12, further comprising:

a register initialization function that is callable by ~~from~~ the threads to specify address extensions.

14. (Original) A multi-tasking operating system as recited in claim 12, wherein each address extension identifies an address range associated with one or more peripheral devices.

15. (Original) One or more computer-readable storage media containing a multi-tasking operating system as recited in claim 12.

16. (Original) A computing device comprising:

a processor that generates a first address signal having a first width;

one or more peripheral devices that are addressed with a second address signal having a second width that is greater than the first width;



PAGE 8/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

an address extension register that stores an address extension, wherein the address extension is combined with the first address signal to create the second address signal; and

the processor being programmed to record the address extension being used by a first execution thread, to interrupt the first execution thread to execute portions of one or more other execution threads, to restore the recorded address extension to the address extension register after executing said portions of one or more other execution threads, and to resume the first execution thread after restoring the recorded address extension to the address extension register.

17. (Original) A computing device as recited in claim 16 wherein each address extension identifies an address range associated with one or more peripheral devices.

18. (Previously presented) A computing device as recited in claim 16 wherein the processor is further programmed to record and restore address extensions for a plurality of execution threads.

19. (Original) A computing device as recited in claim 16 wherein the processor is programmed to perform the following steps upon interrupting the first execution thread to begin execution of the one or more other execution threads:

recording the address extension being used by the one or more other execution threads; and


PAGE 9/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

interrupting the one or more other execution threads to resume the first execution thread.

20. (Previously presented) A computing device comprising:

a processor that generates a first address signal having a first width;

one or more peripheral devices that are addressed with a second address signal having a second width that is greater than the first width;

B1

an address extension register that stores an address extension, wherein the address extension is combined with the first address signal to create the second address signal;

a multi-tasking operating system that switches between execution of different application programs; and

a plurality of application programs that use different address extensions, wherein execution threads of the application programs register one or more such address extensions with the multi-tasking operating system;

wherein the operating system records the registered address extensions and automatically writes an address extension of a particular application program thread to the address extension register before switching to execution of said particular application program thread.

21. (Original) A computing device as recited in claim 20, wherein each address extension identifies an address range associated with one or more peripheral devices.



PAGE 10/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

22. (Previously presented) A computing device as recited in claim 20, wherein the operating system records the registered address extensions in a memory table.

23. (Original) A computing device as recited in claim 20, wherein the execution threads of the application programs invoke operating system device drivers, the operating system device drivers registering said one or more address extensions with the multi-tasking operating system.

24. (Original) A computer program stored in a storage medium, comprising:

instructions for performing read/write operations on a peripheral device, wherein loading an extension register is a predicate to performing said read/write operations; and

instructions for providing one or more address extension values to a multi-tasking operating system for use with one or more threads of the computer program, wherein the operating system automatically loads said one or more address extension values to the extension register whenever switching to said one or more threads of the computer program.

25. (Original) A computer program as recited in claim 24, wherein the address extensions are specified as base addresses.



PAGE 11/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

26. (Previously presented) A computer program stored in a storage medium for execution on a computer, the computer program being configured to cause the computer to performs acts of:

executing an interruptible execution thread of the program;

writing an address extension value associated with the execution thread to an extension register and contemporaneously to a memory location;

associating the memory location with the execution thread;

retrieving the value associated with the execution thread from the memory location when execution of the execution thread is resumed after being interrupted;

writing the value retrieved from the memory location to the extension register; and

resuming application of the execution thread.

27. (Original) A computer program as recited in claim 26, wherein each address extension identifies an address range associated with one or more peripheral devices.

28. (Currently amended) A computer program as recited in claim 26, the computer program being further configured to cause the computer to ~~performs~~ perform acts of alternately executing more than one execution thread.

29. (Currently amended) A computer program as recited in claim 28, the computer program being further configured to cause the computer to ~~performs~~

perform an act of storing more than one address extension value in memory, each value being associated with a particular execution thread.

30. (Currently amended) A computer program as recited in claim 28, the computer program being further configured to cause the computer to ~~performs~~ perform acts of:

storing more than one address extension value in memory, each value being associated with a particular execution thread; and

loading the extension register with the value in memory associated with a particular execution thread prior to resuming execution of that execution thread.

31. (Previously presented) A computer program as recited in claim 28, the computer program being further configured to cause the computer to performs acts of:

storing more than one address extension value in memory, each value being associated with a particular execution thread; and

loading the extension register with the value in memory associated with a particular execution thread prior to resuming execution of that execution thread;

wherein each address extension identifies an address range associated with one or more peripheral devices.

32. (Original) A computer program as recited in claim 28, wherein the computer program executes an operating system device driver from an interruptible execution thread of the program wherein said device driver identifies


PAGE 13/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

a peripheral device to be accessed by the execution thread and identifies a value associated with the execution thread.

33. (Previously presented) A computing device comprising:

a processor that generates a first address signal having a first width;

a first peripheral device addressable with a second address signal having a second width that is greater than the first width;

b1 a second peripheral device addressable with a third address signal having a third width that is greater than the first width;

an address extension register configured to store first and second address extensions, each of which may combined with the first address signal to create the second and third address signals, respectively;

a multi-tasking operating system that switches between execution of different application programs;

a plurality of application programs that use different address extensions, wherein execution threads of the application programs register one or more such address extensions with the multi-tasking operating system;

wherein the operating system records the registered address extensions and automatically writes an address extension of a particular application program thread to the address extension register before switching to execution of said particular application program thread.


PAGE 14/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

34. (Previously presented) A computing device as recited in claim 33, wherein the first and second address extensions identify an address range associated with the respective first and second peripheral devices.

35. (Previously presented) A computing device as recited in claim 33, wherein operating system records the registered address extensions in a memory table.

36. (Previously presented) A computing device as recited in claim 33, wherein the execution threads of the application programs invoke operating system device drivers corresponding to the first and second peripheral devices, the operating system device drivers registering the first and second address extensions with the multi-tasking operating system.



PAGE 15/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

**New Claims:**

37. (New) One or more computer-readable storage media containing computer-readable instructions that, when executed by one or more processors, implement a thread scheduler that schedules multiple execution threads for interleaved execution, wherein the thread scheduler causes an address signal to be generated that is combined with an extended address signal to create a peripheral address signal, wherein the extended address signal is produced from a value stored in an address extension register, the thread scheduler causing acts to be performed comprising:

interrupting a first execution thread to execute portions of one or more other execution threads;

recording the value from the address extension register; and

restoring the recorded value to the extension register after executing said portions of one or more other threads.

38. (New) The computer-readable storage medium of claim 37, wherein the thread scheduler further causes an act of resuming the first execution thread after causing the act of restoring the recorded value to the address extension register to be performed.

39. (New) The computer-readable storage medium of claim 37, wherein the thread scheduler further causes an act of resuming the first execution thread after causing the act of restoring the recorded value to the address extension


PAGE 16/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04

register to be performed, and wherein the thread scheduler further is configured to cause one or more processors to:

record address extension register values associated with a plurality of interrupted execution threads; and

B1

restore the address extension register values to the address extension register when resuming the associated execution threads.



PAGE 17/29 * RCVD AT 3/2/2004 4:46:18 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/0 * DNIS:8729306 * CSID:509 323 8979 * DURATION (mm-ss):07-04